SERVER-GROUP MESSAGING SYSTEM FOR INTERACTIVE APPLICATIONS

Inventors: Daniel Joseph Samuel
Marc Peter Kwiatkowski
Jeffrey Jackiel Rothschild

FIELD OF THE INVENTION

The present invention relates to computer network systems, and particularly to server group messaging systems and methods for reducing message rate and latency.

Background of the Invention

5

There are a wide range of interactive applications implemented on computer systems today. All are characterized by dynamic response to the user. The user provides input to the computer and the application responds quickly. One popular example of interactive applications on personal computers (PCs) are games. In this case, rapid response to the user may mean

10

redrawing the screen with a new picture in between 30ms and 100ms. Interactive applications such as games control the speed of their interaction with the user through an internal time base. The application uses this time base to derive rates at which the user input is sampled, the screen is redrawn and

15

sound is played.

As computers have become more powerful and common, it has become important to connect them together in networks. A network is comprised of nodes and links. The nodes are connected in such a way that there exists a path from each node over the links and through the other nodes to each of the other

5   nodes in the network. Each node may be connected to the network with one or more links. Nodes are further categorized into hosts, gateways and routers. Hosts are computer systems that are connected to the network by one link. They communicate with the other nodes on the network by sending messages and receiving messages. Gateways are computer systems connected to the

10   network by more than one link. They not only communicate with the other nodes as do hosts, but they also forward messages on one of their network links to other nodes on their other network links. This processing of forwarding messages is called routing. In addition to sending and receiving messages and their routing functions, gateways may perform other functions in

15   a network. Routers are nodes that are connected to the network by more than one link and whose sole function is the forwarding of messages on one network link to the other network links to which it is connected. A network consisting of many network links can be thought of as a network of sub-networks with gateways and/or routers connecting the sub-networks together into what is

20   called an internet. Today the widely known example of a world wide internet is the so called "Internet" which in 1995 has over 10 million computers connected full time world-wide.

With so many computers on a single world-wide network, it is desirable to create interactive networked applications that bring together many people in a

25   shared, networked, interactive application. Unfortunately, creating such

shared, networked, interactive applications runs into the limitations of the existing network technology.

As an example, consider a game designed to be deployed over a network which is to be played by multiple players simultaneously. The game could be implemented in software on a PC connected to a network. A rate set by its internal time base, it would sample the inputs of the local user, receive messages from the network from the PCs of the other players and send messages out to the PCs of the other players. A typical rate will be ten time per second for a time period of 100ms. The messages sent between the PCs would contain information that was needed to keep the game consistent between all of the PCs. In a game that created the illusion of a spatial environment where each player could move, the packets could contain information about the new positions of the players as they moved. Today there are many commercial example of PC games that can be played between multiple players on Local Area Networks (LANs) or by two players over dial-up phone lines using modems. The network messages sent by such games contain a wide variety of information specific to the game. This can include position and velocity information of the objects in the game along with special actions taken by a player that effect the other players in the game.

The case of a two player game played over a modem is particularly simple. If the message rate is 10 messages per second, each PC sends 10 messages per second to the other PC and receives 10 messages per second. The delay introduced by the modems and phone line is small and will not be noticed in most games. Unfortunately, the case of two players is uninteresting for networked interactive applications. With the same game played with 8 players on a LAN, the message rate increases. Each PC must send 7 messages, one to

each of the other 7 players every time period and will receive 7 messages from the other players in the same time period. If the messaging time period is 100ms, the total message rate will be 70 messages sent per second and 70 messages received per second. As can be seen the message rate increases linearly with the number of players in the game. The message rates and data rates supported by popular LANs are high enough to support a large number of players at reasonable message sizes. Unfortunately, LANs are only deployed in commercial applications and cannot be considered for deploying a networked interactive application to consumer users.

The wide area networks available today to consumer users all must be accessed through dial-up phone lines using modems. While modem speeds have increased rapidly, they have now reached a bit rate of 28.8 Kbits/sec which is close to the limit set by the signal-to-noise ratio of conventional phone lines. Further speed increases are possible with ISDN, but this technology is not ready for mass market use. Other new wide area networking technologies are being discussed that would provide much higher bandwidth, but none are close to commercial operation. Therefore, in deploying a networked, interactive application to consumers, it is necessary to do so in a way that operates with existing networking and communications infrastructures.

In the example of the 8 player networked game, consider a wide area network implementation where the PCs of each of the players is connected to the network with a 28.8 Kbit/sec modem. Assume that the network used in this example is the Internet so that all of the network protocols and routing behavior is well defined and understood. If the game uses TCP/IP to send its messages between the PCs in the game, the PPP protocol over the dial-up phone lines can be advantageously used to compress the TCP/IP headers.

Even so, a typical message will be approximately 25 bytes in size. Sent through the modem, this is 250 bits. The messages are sent 10 times per second to each of the other PCs in the game and received 10 times per second from the other PCs. This is 35.0 Kbits/sec which exceeds the capabilities of the modem by 20%. If the messages are reduced to 20 bytes, just 8 players can be supported, but this approach clearly cannot support networked interactive applications with large numbers of participants. There are other problems beyond just the bandwidth of the network connection. There is the loading on each PC caused by the high packet rates and there is the latency introduced by the time needed to send all of the outbound packets. Each packet sent or received by a PC will require some amount of processing time. As the packet rate increases with the number of players in the game, less and less of the processor will be available for running the game software itself. Latency is important in an interactive application because it defines the responsiveness of the system. When a player provides a new input on their system, it is desirable for that input to immediately affect the game on all of the other players systems. This is particularly important in any game where the game outcome depends on players shooting at targets that are moved by the actions of the other players. Latency in this case will be the time from when a player acts to move a target to the time that the target has moved on the screens of the other players in the game. A major portion of this latency will come from the time needed to send the messages to the other seven players in the game. In this example the time to send the messages to the other 7 players will be approximately 50 ms. While the first player of the seven will receive the message quickly, it will not be until 50 ms have passed that the last player of the seven will have received the message.

Internet Protocol Multicasting

As mentioned before, the Internet is a widely known example of a wide area network. The Internet is based on a protocol appropriately called the Internet Protocol (IP). In the OSI reference model for layers of network

5 protocols, IP corresponds to a layer 3 or Network layer protocol. It provides services for transmission and routing of packets between two nodes in an internet. The addressing model provides a 32 bit address for all nodes in the network and all packets carry source and destination addresses. IP also defines the routing of packets between network links in an inter-network. Gateways

10 and routers maintain tables that are used to lookup routing information based on the destination addresses of the packets they receive. The routing information tells the gateway/router whether the destination of the packet is directly reachable on a local network link connected to the gateway/router or if not, the address of another gateway/router on one of the local network links to

15 which the packet should be forwarded. On top of IP are the layer 4 transport protocols TCP and UDP. UDP provides datagram delivery services to applications that does not guarantee reliable or in-order delivery of the datagrams. TCP is a connection oriented service to applications that does provide reliable delivery of a data stream. It handles division of the stream into

20 packets and ensures reliable, in-order delivery. See the Internet Society RFCs: RFC-791 "Internet Protocol", RFC-793 "Transmission Control Protocol" and RFC-1180 "A TCP/IP Tutorial". IP, TCP and UDP are unicast protocols: packets, streams or datagrams are transmitted from a source to a single destination.

25 As an example, consider Figures 1 and 2. Figure 1 shows a conventional unicast network with hosts 1, 2, 3 and 4 and network links 11, 12, 13, 14,

15,16,17, 18 and 19 and routers 5, 6, 7, 8, 9 and 10. In this example, each host wants to send a data payload to each of the other hosts. Host 1 has network address A, host 2 has network address C, host 3 has network address B and host 4 has network address D. Existing network protocols are typically based on packet formats that contain a source address, destination address and a payload. This is representative of commonly used wide area network protocols such as IP. There are other components in an actual IP packet, but for sake of this example, only these items will be considered. Figure 2 shows the example packets that are sent by the hosts to one another using a conventional unicast network protocol such as IP. Host 1 send packets 20, to host 3, packet 21 to host 2 and packet 22 to host 4. Host 1 wants to send the same data P1 to each of the other three hosts, therefore the payload in all three packets is the same. Packet 20 travels over network links 11, 12, 15 and 18 and through routers 5, 6, and 8 to reach host 3. In a similar fashion host 3 sends packets 23 to host 1, packet 24 to host 2 and packet 25 to host 4. Host 2 and host 4 send packets 26, 27, 28 and 29, 30, 31 respectively to the other three hosts. All of these packets are carried by the unicast network individually from the source host to the destination host. So in this example each host must send three packets and receive three packets in order for each host to send its payload to the other three hosts.

As can be seen, each host must send a packet to every other host that it wishes to communicate with in an interactive application. Further, it receives a packet from every other host that wishes to communicate with it. In an interactive application, this will happen at a regular and high rate. All of the hosts that wish to communicate with one another will need to send packets to each other eight to ten times per second. With four hosts communicating with

one another as in this example, each host will send three messages and receive three messages eight to ten times per second. As the number of hosts in the application that need to communicate with one another grows, the message rate will reach a rate that cannot be supported by conventional dial-up lines.

5 This makes unicast transport protocols unsuitable for delivering interactive applications for multiple participants since their use will result in the problem of high packet rates that grow with the number of participants.

Work has been done to attempt to extend the IP protocol to support

10 multicasting. See RFC-1112 "Host Extensions for IP Multicasting.". This document describes a set of extensions to the IP protocol that enable IP multicasting. IP multicasting supports the transmission of a IP datagram to a host group by addressing the datagram to a single destination address. Multicast addresses are a subset of the IP address space and identified by class

15 D IP addresses - these are IP addresses with "1110" in the high order 4 bits. The host group contains zero or more IP hosts and the IP multicasting protocol transmits a multicast datagram to all members of the group to which it is addressed. Hosts may join and leave groups dynamically and the routing of multicast datagrams is supported by multicast routers and gateways. It is

20 proper to describe this general approach to multicast messaging as "distributed multicast messaging". It is a distributed technique because the job of message delivery and duplication is distributed throughout the network to all of the multicast routers. For distributed multicast messaging to work in a wide area network, all of the routers handling datagrams for multicast hosts must support

25 the routing of multicast datagrams. Such multicast routers must be aware of the multicast group membership of all of the hosts locally connected to the

router in order to deliver multicast datagrams to local hosts. Multicast routers must also be able to forward multicast packets to routers on their local network links. Multicast routers must also decide to which if any local routers they must forward multicast datagrams. When a multicast datagram is received, by

5 a multicast router, its group address is compared to a list for each local multicast router of group addresses. When there is a match, the datagram is then forwarded to that local multicast router. Therefore, the multicast routers in the network must maintain an accurate and up to date list of group addresses for which they are to forward datagrams to. These lists are updated when

10 hosts join or leave multicast groups. Hosts do this by sending messages using Internet Group Management Protocol (IGMP) to their immediately-neighboring multicast routers. A further attribute of distributed multicast messaging is that the routers must propagate the group membership information for a particular group throughout the network to all of the other

15 routers that will be forwarding traffic for that group. RFC-1112 does not describe how this is to be done. Many different approaches have been defined for solving this problem that will be mentioned later in descriptions of related prior art. Despite their differences, all of these approaches are methods for propagation of multicast routing information between the multicast routers and

20 techniques for routing the multicast datagrams in an inter-network supporting distributed multicast messaging.

The distributed multicast messaging approach has a number of undesirable side effects. The process of propagation of group membership information to all of the relevant routers is not instantaneous. In a large complex network it

25 can even take quite a period of time depending on the number of routers that must receive that updated group membership information and how many

routers the information for the group membership update must past through. This process can easily take many seconds and even minutes depending on the specifics of the algorithm that is used. RFC-1112 mentions this problem and some of the side effects that must be handled by an implementation of a

5   practical routing algorithm for multicast messaging. One problem results when groups are dynamically created and destroyed. Since there is no central authority in the network for assigning group addresses, it is easily possible in a distributed network for there to be duplication of group address assignment. This will result in incorrect datagram delivery, where hosts will receive

10 unwanted datagrams from the duplicate group. This requires a method at each host to filter out the unwanted datagrams. Another set of problems result from the time delay from when a group is created, destroyed or its membership changed to when all of the routers needed to route the datagrams to the member hosts have been informed of these changes. Imagine the case where

15 Host N joins an existing group by sending a join message to its local router. The group already contains Host M which is a number of router hops away from Host N in the network. Shortly after Host N has sent it join message, Host M sends a datagram to the group, but the local router of Host M has not yet been informed of the change in group membership and as a result the

20 datagram is not forwarded to one of the particular network links connected to the local router of Host M that is the only path in the network from that router that ultimately will reach Host N. The result is that Host N will receive no datagrams addressed to the group from Host M until the local router of M has its group membership information updated. Other related problems can also

25 occur. When a host leaves a group, messages addressed to the group will continue for some time to be routed to that host up to the local router of that

host. The local router will know at least not to route the datagram onto the local network of that host. This can still result in a great deal of unnecessary datagrams being carried in a large network when there are many active message groups with rapidly changing memberships.

Finally, distributed multicast messaging does not sufficiently reduce the message rate between the hosts. With distributed multicast messaging, each host need only send one message addressed to the message group in order to send a message to all of other hosts in the group. This is an improvement over conventional unicast messaging where one message would need to be sent to each of the other hosts in a group. However, distributed multicast messaging does nothing to reduce the received message rate at each of the hosts when multiple hosts in a group are sending messages to the group closely spaced in time. Let us return to the example of a group of ten hosts sending messages seven times per-second to the group. With conventional unicast messaging, each host will need to send 9 messages to the other hosts, seven times per-second and will receive 9 messages, seven times per-second. With distributed multicast messaging, each host will need to send only one message to the group containing all of the hosts seven times per-second, but will still receive 9 messages, seven times per-second. It is desirable to further reduce the number of received messages.

An example of distributed multicasting is shown in Figures 3 and 4. Figure 3 shows a network with multicast routers 39, 40, 41, 42, 43 and 44 and hosts 35, 36, 37, 38 and network links 45, 46, 47, 48, 49, 50, 51, 52 and 53. The four hosts have unicast network addresses **A**, **B**, **C**, **D** and are also all members of a message group with address **E**. In advance the message group was created and each of the hosts joined the message group so that each of the multicast

routers is aware of the message group and has the proper routing information. A network protocol such IP with multicast extensions is assumed to be used in this example. Host 35 sends packet 54 with source address **A** and destination multicast address **E** to the entire message group. In the same manner host 37

5 sends packet 55 to the group, host 36 sends packet 56 to the group and host 38 sends packet 57 to the group. As the packets are handled by the multicast routers they are replicated as necessary in order to deliver them to all the members of the group. Let us consider how a packets sent by host 35 is ultimately delivered to the other hosts. Packet 54 is carried over network link

10 45 to multicast router 39. The router determines from its routing tables that the multicast packet should be sent onto network links 46 and 47 and duplicates the packet and sends to both of these network links. The packet is received by multicast routers 40 and 43. Multicast router 43 sends the packet onto network link 50 and router 40 sends its onto links 48 and 49. The packet

15 is then received at multicast routers 44, 42 and 41. Router 41 sends the packet over network link 51 where it is received by host 36. Router 42 sends the packet over network link 52 to host 37 and router 44 sends the packet over link 53 to host 38. A similar process is followed for each of the other packets sent by the hosts to the multicast group **E**. The final packets received by each

20 host are shown in Figure 4.

While distributed multicasting does reduce the number of messages that need to be sent by the hosts in a networked interactive application, it has no effect on the number of messages that they receive. It has the further disadvantages of poor behavior when group membership is rapidly changing

25 and requires a special network infrastructure of multicast routers. It also has no support for message aggregation and cannot do so since message delivery is

distributed. Distributed multicasting also has no support for messages that define logical operations between message groups and unicast host addresses.

All of these problems can be understood when placed in context of the design goals for distributed multicast messaging. Distributed multicast
5   messaging was not designed for interactive applications where groups are rapidly created, changed and destroyed. Instead it was optimized for applications where the groups are created, changed and destroyed over relatively long time spans perhaps measured in many minutes or even hours. An example would be a video conference where all the participants agreed to
10   connect the conference at a particular time for a conference that might last for an hour. Another would be the transmission of an audio or video program from one host to many receiving hosts, perhaps measured in the thousands or even millions. The multicast group would exist for the duration of the audio/video program. Host members would join and leave dynamically, but in
15   this application it would be acceptable for there to be a significant time lag from joining or leaving before the connection was established or broken.

While IP and multicast extensions to IP are based on the routing of packets, another form of wide area networking technology called Asynchronous Transfer Mode (ATM) is based on switching fixed sized cells through switches.
20   Unlike IP which supports both datagram and connection oriented services, ATM is fundamentally connection oriented. An ATM network consists of ATM switches interconnected by point-to-point links. The host systems are connected to the leaves of the network. Before any communication can occur between the hosts through the network, a virtual circuit must be setup across
25   the network. Two forms of communication can be supported by an ATM network. Bi-directional point-to-point between two hosts and point-to-

14

multipoint in one direction from one host to multiple hosts. ATM, however, does not directly support any form of multicasting. There are a number of proposals for layering multicasting on top of ATM. One approach is called a multicast server, shown in Figure 8. Host systems 112, 113, 114, 115 setup

5 point-to-point connections 106, 107,108 and 109 to a multicast server 105. ATM cells are sent by the hosts to the multicast server via these links. The multicast server sets up a point-to-multipoint connection 111 to the hosts which collectively constitute a message group. Cells sent to the server which are addressed to the group are forwarded to the point-to-multipoint link 111.

10 The ATM network 110 is responsible for the transport and switching for maintaining all of the connections between the hosts and the server. The cells carried by the point-to-multipoint connection are duplicated when necessary by the ATM switches at the branching points in the network tree between and forwarded down the branching network links. Therefore, the network is

15 responsible for the replication of the cells and their payloads, not the server. This method has the same problems as distributed multicasting when used for an interactive application. Each host still receives individual cells from each of the other hosts, so there is no aggregation of the payloads of the cells targeted at a single host. There is no support for addressing cells to hosts based on

20 logical operations on the sets of members of host groups.

Related Prior Art

There are a number of existing patents and European patent applications that are related to the area of the invention. These can be organized into two separate categories: multicast routing/distribution and source to destination

25 multicast streams.

Multicast routing and distribution

These patents are US 4,740,954 by Cotton et al, US 4,864,559 by Perlman, US 5,361,256 by Doeringer et al, US 5,079,767 by Perlman and US 5,309,433 by Cidon et al. Collectively these patents cover various algorithms for the routing and distribution of the datagrams in distributed multicast networks. None deal with the problems described previously for this class of multicast routing and message distribution such as poor behaviors when the message groups change rapidly. In all of these patents, messages are transmitted from a host via a distributed network of routers to a plurality of destination hosts which are members of a group. Since these patents deal only with variants of distributed multicasting they provide no means to reduce the received message rate, no method to aggregate messages and provide no method in the messages to perform logical operation on message groups.

Source to destination multicast streams

These are PCTs and a European patent application. They are EP 0 637 149 A2 by Perlman et al, PCT/US94/11282 by Danneels et al and PCT/US94/11278 by Sivakumar et al. These three patent applications deal with the transmission of data streams from a source to a group of destinations. In none of these patent applications, is a method described for transmitting data between multiple members of a group. In all of these applications, the data transmission is from a source to a plurality of designations. Since these patent applications deal only with point-to-multipoint messaging, they can provide no means to reduce the received message rate, no method to aggregate messages and provide no method in the messages to perform logical operation on message groups.

SUMMARY OF THE INVENTION

The present invention relates to facilitating efficient communications between multiple host computers over a conventional wide area communications network to implement an interactive application such as a

5   computer game between multiple players. In such an application, the hosts will be dynamically sending to each other information that the other hosts need in order to keep the interactive application operating consistently on each of the hosts. The invention is comprised of a group messaging server connected to the network that maintains a set of message groups used by the hosts to

10   communicate information between themselves. The invention further comprises a server-group messaging protocol used by the hosts and the server. The server-group messaging protocol is layered on top of the Transport Level Protocol (TLP) of the network and is called the Upper Level Protocol (or ULP). In the OSI reference model the ULP can be thought of as a session

15   layer protocol built on top of a transport or applications layer protocol. The ULP protocol uses a server-group address space that is separate from the address space of the TLP. Hosts send messages to addresses in the ULP address space to a group messaging server using the underlying unicast transport protocol of the network. The ULP address space is segmented into

20   unicast addresses, implicit group messaging addresses and logical group messaging addresses. The implicit and logical group messaging addresses are collectively called group messaging addresses.

Host systems must first establish connections to a group messaging server before sending messages to any ULP addresses. The process of establishing

25   this connection is done by sending TLP messages to the server. The server establishes the connection by assigning a unicast ULP address to the host and

returning this address in an acknowledgment message to the host. Once connected, hosts can inquire about existing message groups, join existing message groups, create new message groups, leave message groups they have joined and send messages to ULP addresses known by the server. Each

5   message group is assigned either an implicit or logical ULP address depending on its type.

Figure 5 shows an example of a wide area network with a group messaging server ("GMS"). Hosts 58 has TLP address **A** and ULP address **H**, host 59 has TLP address **C** and ULP address **J**, host 60 has TLP address **B** and ULP

10   address **I** and host 61 has TLP address **D** and ULP address **K**. The network is a conventional unicast network of network links 69, 70, 71, 72, 73, 74, 75, 76, and 77 and unicast routers 63, 64, 65, 66, 67, and 68. The group messaging server 62 receives messages from the hosts addressed to a message group and send the contents of the messages to the members of the message group.

15   Figure 6 shows an example of datagrams sent from the hosts to a message group that they are members of. As before, a TLP such as IP (where the message header contain the source and destination TLP addresses) is assumed to be used here. Host 58 sends message 80 which contains the TLP source address **A** of the host and the destination TLP address **S** for the GMS 62. The

20   destination ULP address **G** is an implicit ULP address handled by the GMS and the payload P1 contains both the data to be sent and the source ULP address **H** of the host. It is assumed that prior to sending their ULP messages to the GMS, that each host as already established a connection to the GMS and joined the message group **G**. Host 60 sends message 81 with payload P2

25   containing data and source ULP address **I**. Hosts 59 sends message 82 with payload P3 containing data and source ULP address **J**. Host 61 sends message

83 with payload P4 containing data and source ULP address **K**. The GMS receives all of these messages and sees that each message is addressed to implicit message group **G** with members **H**, **I**, **J**, and **K**. The GMS can either process the message with or without aggregating their payloads. Figure 6

5 shows the case where there is no aggregation and Figure 7 shows the case with aggregation.

Without aggregation, the GMS generates the outbound messages 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, and 95 which it sends to the hosts. The datagrams have TLP headers with the source and destination TLP addresses of

10 the GMS and the hosts respectively. The next field in the datagrams is the destination ULP of the datagram. Datagrams 84, 85, and sent to host 58 with TLP address **A** and ULP address **H**. Datagrams 87, 88, and 89 are sent to host 60 with TLP address **B** and ULP address **I**. Datagrams 90, 91 and 92 are sent to host 59 with TLP address **C** and ULP address **J**. Datagrams 93, 94 and 95

15 are sent to host 61 with TLP address **D** and ULP address **K** respectively. As can be seen from the payloads that each host has received, each host has received the payloads from the other three hosts. Note that each host has not received a copy of its own original message. This is because the GMS has performed echo suppression. This is selectable attribute of the GMS since in

20 some applications it is useful for the hosts to receive and echo of each message that they send to a group that they are also members of. In the example of Figure 6, it has been shown how the present invention can achieve the same message delivery as distributed multicasting without its disadvantages. Without aggregation, the present invention enables a host to send a single

25 message to multiple other hosts that are members of a message group. It reduces the message traffic that a host must process in an interactive

19

application by reducing the number of messages that each host must send to the others. Without aggregation, however, there is no reduction in the number of messages received by the hosts. Without aggregation we can achieve the same message rate as distributed multicasting without the need for a network with

5   multicast routers, we can use a conventional unicast network such as the Internet. The present invention also avoids the problems that dynamic group membership causes for distributed multicasting. Group membership can be changed very rapidly. Groups can be created, joined and left by single unicast messages from hosts to the GMS. These messages will be point-to-point

10   messages and will not have to propagate in throughout the network nor have to cause routing table changes in the routers. This ability to rapidly and accurately change group membership is critical to the implementation of networked interactive applications. Consider a computer game for multiple players that supports hundreds of players that are spread throughout a three

15   dimensional space created by the game. At any time only a few players will be able to see and effect one another in the game since other players will be in other areas that are out of sight. Using conventional phone lines to carry the data from each players computer to the network, it will not be possible to send all actions of each player to all of the other players, but because only a few

20   players will be in close proximity at any one time, it will not be necessary to do so. It is only necessary to send data between the players that are in close proximity to one another. These "groups" of players naturally map onto the message groups of the invention. As players move about the three dimensional space of the game, game will cause them to join and leave message groups as

25   necessary. If this does not happen rapidly it will limit the interactivity of the game or cause inconsistent results for the different players in the game.

The invention also allows aggregating message payloads of multiple messages destined to a single host into a single larger message. This can be done because of the GMS where all of the messages are received prior to being sent to the hosts. Figure 7 shows an example of how this works. The hosts send their messages to the GMS in exactly the same fashion as in Figure 6 using the same addresses previously defined in Figure 5. Host 58 sends message 96, host 60 sends message 97, host 59 sends message 98 and host 61 sends message 99. The GMS receives all of these messages and creates four outbound messages 100, 101, 102 and 103. The process by which these messages will be explained in detail in the detailed description of the invention. Each message is destined to a single host and contains an aggregated payload with multiple payload items. Message 100 has a destination ULP address **H** for host 58 and aggregated payload P2, P3 and P4 from the messages from hosts 59, 60 and 61. Message 101 is targeted at host 60, message 102 is targeted at host 59 and message 103 is targeted at host 61. As can be seen, each host sends one message and receives one message. The received message is longer and contains multiple payloads, but this is a significant improvement over receiving multiple messages with the wasted overhead of multiple message headers and message processing time. Overall the invention has dramatically reduced the amount of data that must be sent and received by each host. Since the bit rate over conventional phone lines using a modem is low, a reduction in the amount of data that must be sent and received directly translates into improved time and latency for message communications between the hosts.

Hosts create, join and leave message groups using control messages in the ULP protocol to the GMS. Hosts may also read and write application specific state information that is stored in the GMS. When hosts send messages to

other hosts, the message must be at least addressed to an implicit group address. The ULP implicit address will always be the primary address in a message from one host to another. The message may optionally specify auxiliary destination addresses. In many cases the implicit ULP address will be

5   the only destination ULP address in the message. The GMS will handle delivery of the ULP messages addressed to the implicit message group to all of the hosts that are members of the group. A ULP send message may optionally specify an address list of auxiliary addresses in addition to the primary destination of the implicit ULP address. This auxiliary address list can contain

10   only unicast and logical ULP addresses. The address list can also specify set operators to be performed between the sets of host ULP addresses defined by the unicast addresses and logical groups. Once the address list has been processed to yield a set of hosts, this set is intersected with the set of hosts that are members of the implicit message group specified by the primary implicit

15   ULP address in the message. This ability to perform logical set operators on message groups is very useful in interactive applications. It allows a single ULP message to selectively deliver a message to hosts that fit a set of computed criteria without the sending host having to know the anything about the members of the groups in the address list. Recall the example of a

20   networked game with hundreds of players in a three dimensional environment created by the game. Consider an implicit message group consisting of all of the game players in a certain area of the game where all of the players can interact with one another. Consider that the players are organized into multiple teams. Logical message groups could be created for each team within the

25   game. To send a message to all the players within the area that were on one team, a ULP message would be sent to the ULP implicit message group for all

the players in the area with an auxiliary address of the logical message group for all the players on the selected team. The GMS would perform the proper set intersection prior to sending the resulting messages to the targeted hosts. The result of this will be that the message will only be delivered to the players

5   on the selected team in the selected area of the game.

In summary, the present invention deals with the issues of deploying an interactive application for multiple participants on wide area networks by providing a method for reducing the overall message rate and reducing latency. This invention uses a server group messaging approach, as oppose to the above

10   described "distributed multicast messaging" approach. The present invention overcomes the undesirable side effects of the distributed multicast messaging approach. Further, it reduces the message rate between the hosts. As pointed out in an example discussed above, with prior art distributed multicast messaging, each host will need to send only one message to the group

15   containing all of the hosts seven times per-second, but will still receive 9 messages, seven times per-second. The present invention of server group messaging has each host sending one message, seven times per-second and receiving one message, seven times per-second.

The present invention is different from the multicast routing and

20   distribution method disclosed in U.S. Patent Nos. 4,740,954, 4,864,559, 5,361,256, 5,079,767 and 5,309,433. Since these patents deal only with variants of distributed multicasting they provide no means to reduce the received message rate, no method to aggregate messages and provide no method in the messages to perform logical operation on message groups. This

25   differs from the present invention where messages from multiple hosts

addressed to a message group are received by a group server which processes the contents of the messages and transmits the results to the destination hosts.

The present invention is also different from the source to destination multicast streams approach disclosed in EP 0 637 149 A2, PCT/US94/11282 and PCT/US94/11278. In all of these references, the data transmission is from a source to a plurality of designations, whereas the present invention describes data transmission from a sending host to a server host system and then from the server host to the destination hosts.

These and other features and advantages of the present invention can be understood from the following detailed description of the invention together with the accompanying drawings.

DESCRIPTION OF DRAWINGS

Figure 1 shows a conventional unicast network consisting of hosts, network links and routers.

Figure 2 shows the unicast datagrams on a conventional unicast network that would be needed to implement an interactive application between four hosts.

Figure 3 shows a prior art multicast network consisting of hosts, network links and multicast routers.

Figure 4 shows a multicast datagrams on a prior art multicast network that would be needed to implement an interactive application between four hosts.

Figure 5 shows a unicast network equipped with a group messaging server in accordance with the present invention.

Figure 6 shows the ULP datagrams without payload aggregation on a network according to the present invention that would be needed to implement an interactive application between four hosts.

Figure 7 shows the ULP datagrams with payload aggregation on a network according to the present invention that would be needed to implement an interactive application between four hosts.

Figure 8 shows a prior art ATM network with a multicast server.

5 Figure 9 shows the detailed datagram format and address format for ULP messages in accordance with the present invention.

Figure 10 shows the internal functions of the GMS according to the present invention.

Figure 11 shows the host software interface and functions needed to

10 support the ULP according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method for multiple host computers to efficiently communicate information to one another over a wide area network for the purposes of implementing an interactive application between multiple

15 users. The method consists of three components: a host protocol interface, a protocol and a server. The protocol is between the host protocol interface and the server and is implemented on top of the network transport protocol of a wide area network. The protocol is called the Upper Level Protocol (ULP) since it is layered above the existing network Transport Level Protocol (TLP).

20 In the OSI reference model the protocol can be described as a Session Layer protocol on top of the Transport Layer of the network. Figure 11 shows the host protocol interface, 151, relative to the interactive application, 150, and the host interface for the Transport Level Protocol , 153. The network interface, 155, provides the physical connection for the host to the network. The

25 network communications stack, 154, is the communications protocol stack that provides network transport services for the host and the host interface for the

Transport Level Protocol, 153, is and interface between host application software and the network transport services of the network communications stack.

The interactive application can send and receive conventional network messages using the host interface to the TLP. The interactive application also can send and receive ULP messages through the host interface for the ULP. Internal to the host interface for the ULP is a table, 152, of all ULP addresses which the host can send messages to. Each entry in the table contains a pair of addresses, a ULP address and its corresponding TLP address. When the host sends a message to a ULP address, that message is encapsulated in a TLP message sent to the TLP address corresponding to that ULP address. This allows the ULP messages to be handled transparently by the transport mechanisms of the existing network. A core function of the ULP is group messaging where hosts send messages to message groups populated by multiple hosts. This allows a host to send a message to multiple hosts with one ULP message. Since the ULP is layered on top of the TLP, the group messaging functions of the ULP operate on a conventional unicast network where TLP messages can only be sent from one host to only one other host.

The group based messaging is implemented through the use of a server called a group messaging server. All ULP messages from the hosts are sent from the hosts to a group messaging server using the TLP protocol. The server processes the ULP portion of the messages and takes the necessary required by the ULP message. Control ULP messages are processed locally by the server and may be acknowledged to the sending host. ULP messages addressed to other hosts are processed by the group messaging server and then

re-transmitted to the proper ULP destination hosts, again using the TLP protocol to encapsulate and transport these messages.

In Figure 5, hosts 58, 59, 60 and 61 send messages to one another using the ULP over a conventional unicast network using a group messaging server 62. The network consists of conventional routers 63, 64, 65, 66, 67 and 68 connected with conventional network links 69, 70, 71, 72, 73, 74, 75, 76 and 77. Host 58 can send a message to hosts 59, 60 and 61 by sending a single ULP message to the group messaging server 62 where the ULP message specifies a destination address that is a ULP message group. The ULP message is encapsulated in a TLP message addressed to the group messaging server. This causes the message to be properly routed by router 63 to network link 71 to router 67 to the server 62. The group messaging server receives the ULP message and determines that the message is addressed to a message group containing hosts 59, 60 and 61 as members. The server sends the payload of the received message to each of the hosts in three new ULP messages individually sent to the three hosts. Since each message is encapsulated in a TLP message, the messages are properly carried over the conventional unicast network. The first ULP message is sent by the group messaging server to host 61. This message is carried by network links 71, 70, 72 and 75 and routers 67, 63, 64 and 65. The second ULP message is sent by the group messaging server to host 60. This message is carried by network links 71, 70, 73 and 76 and routers 67, 63, 64 and 66. The third ULP message is sent by the group messaging server to host 61. This message is carried by network links 74 and 77 and routers 67 and 68.

27

The invention can be implemented both in a datagram form and in a connection oriented form. To best understand the details of the invention, it is best to first consider a datagram implementation.

Datagram Transport Implementation

The ULP can be implemented as a datagram protocol by encapsulating addresses, message type information and the message payload within a datagram of the underlying network transport protocol. The general form of the ULP datagram message format is shown in Figure 9 as elements 123, 124, 125, 126, 127, 128 and 129. The transport header 123 is the datagram header of the TLP that is encapsulating the ULP datagram. The ULP message type field 124 indicates whether it is a send or receive message, if it is a control message or a state message. The following table shows the different message types. The ULP message type field must be present in a ULP datagram.

Message Types	
1	Send
2	Receive
3	Send Control
4	Receive Control
5	Send State
6	Receive State

Send messages are always sent from a host to a group messaging server. Messages from a group server to the hosts are always receive messages. Send Control messages are messages from hosts to a group messaging server requesting a control function be performed. Receive Control messages are acknowledgments from a group messaging server to the hosts in response to a

prior Send Control messages. The Send and Receive State messages are special cases of the Send and Receive Control messages that allow hosts to read and write application specific state storage in the group messaging server. The specific control functions supported by the ULP will be explained later.

5   The destination ULP address 125 is required in ULP datagrams and specifies the primary destination of the ULP message. The address count field 126 is required in ULP send message types and is not present in ULP receive message types. When the address count field in a ULP send message is non-zero, it specifies the number of auxiliary destination addresses for the send

10   message that follow the address count field. These auxiliary destination addresses are shown as items 127 and 128, but it is understood that there are as many auxiliary ULP destination addresses as specified by the address count field. Finally there is the payload 129.

The payload format for ULP datagrams is defined by items 116, 117, 118,

15   119, 120, 121 and 122. Item 116 is the message count and defines how many payload elements will be contained in the payload. A single payload element consists of a triplet of source ULP address, data length and data. Items 117, 118 and 119 comprise the first payload element of the payload. Item 117 is the ULP address of the source of the payload element, item 118 is the data length

20   for the data in the payload element and item 119 is the actual data. Items 120, 121 and 122 comprise the last payload element in the payload. ULP send messages only support payloads with a single payload element, so the message count is required to be equal to one. ULP receive messages may have payloads with one or more payload elements.

29

ULP Address Space

The address space of the ULP is divided into three segments: unicast host addresses, implicit group addresses and logical group addresses. All source and destination addresses in ULP must be in this address space. The ULP

5   address space is unique to a single group messaging server. Therefore each group messaging server has a unique ULP address space. Multiple group messaging servers may be connected to the network and hosts may communicate with multiple group messaging servers without confusion since each ULP datagram contains the header of the TLP. Different group

10   messaging servers will have unique TLP addresses which can be used by the hosts to uniquely identify multiple ULP address spaces. The format for ULP addresses is shown in Figure 9 comprised of items 130, 131 and 132. The address format field 130 is a variable length field used to allow multiple address lengths to be supported. The address type field 131 indicates the type of ULP

15   address: unicast host, implicit group or logical group. The encoding is as follows:

Address Type Encoding	
0 0	Unicast Host Address
0 1	Unicast Host Address
1 0	Implicit Group Address
1 1	Logical Group Address

The address format encoding determines the length of the address field and

25   therefore the total length of the ULP address. This encoding is shown below. Note that when the address type specifies a unicast host address, the low bit of the address type field is concatenated to the address field to become the most significant bit of the address. This doubles the size of the address space for

30

unicast host addresses which is useful since there will generally be more hosts than group messaging servers.

Address Format	Encoding
0	29 Bit Address Field
1 0	4 Bit Address Field
1 1 0	11 Bit Address Field

ULP unicast host addresses are assigned to each host when it first connects to a group messaging server. When a host sends a message to other ULP address, the unicast ULP address of the host will appear as the source ULP address in the received payload element. Unicast ULP host addresses can also be used as destination addresses only as auxiliary addresses in a ULP send message. They are not allowed to be used to as the primary ULP destination address. This means that hosts cannot send ULP directly to one another, but always must send the messages to one another through a group messaging server.

Implicit group addresses are created by a group messaging server in response to a control message to the server requesting the creation of an implicit message group. The host requesting the creation of the implicit message group becomes a member of the message group when it is created. Other hosts can send inquiry control messages to the group messaging server to learn of its existence and then send a implicit group join message in order to join the group. The group messaging server maintains a list of ULP addresses of hosts that are members of the implicit message group. Implicit ULP group addresses are the only ULP addresses allowed to be the primary destination of a ULP send message. Implicit ULP addresses will never appear as ULP source addresses in a payload element.

Logical ULP addresses are used both to address logical message groups and for specifying set operations between the group members of the auxiliary ULP addresses in a ULP send message. Logical message groups are created and joined similarly to implicit message groups, however, logical ULP

5   addresses may only be used as auxiliary ULP addresses in a ULP send message. Logical ULP addresses will also never appear as source ULP addresses in a payload element. The support of set operations between message groups as part of a ULP send message will be explained in a later section on ULP send messages.

10   **Group Messaging Server Internal Functions**

The internal components of the group messaging server are shown in Figure 10.

In the preferred embodiment, the group messaging server is a general purpose computer system with a network interface to connect it to a wide area

15   network. Item 135 is the network interface for the group messaging server and includes not only the hardware connection to the network but the communications protocol stack used to implement the TLP on the server.

Item 136 is an overall control function for the group messaging server. This control function is responsible for all ULP messages that are sent or

20   received by the GMS. Internal to this control function are several important storage and processing functions. Item 137 is an address map for all hosts currently connected to the GMS. This address map is a list of the ULP host address of each host connected to GMS and its corresponding TLP address. This enables the control function to construct the necessary TLP headers for

25   sending ULP messages to the hosts connected to the GMS. Item 138 is a list of all of the currently active implicit ULP addresses currently recognized by the

GMS. Item 139 is an application specific state storage and processing function. Many interactive applications deployed over a network will be able to be implemented solely with host based processing. In these cases all data that needs to be sent between the hosts can be transported using the ULP.

5   However, some applications will need maintain a centrally stored and maintained repository of application state information. This is useful when hosts may join or leave the application dynamically. When hosts join such an application, they will need a place from which they can obtain a snapshot of the current state of the application in order to be consistent with the other hosts

10   that already where part of the application. To read and write this state storage area, the ULP supports send and receive state message types. Within these messages, there is the ability to access a state address space so that different portions of the state can be individually accessed. Application specific processing of state written into this state storage area can also be implemented.

15   Items 140 and 141 are two of multiple ULP server processes running on the GMS. These are software processes that are at the heart of the ULP. Each implicit ULP addresses recognized by the GMS has a one-to-one correspondence to a ULP server process and to a message group maintained by the process. Since all ULP send messages must have an implicit ULP address

20   as the primary destination address of the message, every ULP send message is sent to and processed by a ULP server process. These processes are created by the GMS control function in response to ULP control messages to create new implicit ULP addresses. They are destroyed when the last host which is a member of its message group has left the message group. Internal to a ULP

25   server process is a list, 142, of the ULP host addresses of the members of the message group, a set of message queues 143 for each host which is a member

of the message group and a message aggregation function 149 which is used to aggregate multiple messages to a single host into a single message.

Item 145 maintains a list of all of the logical ULP addresses and message groups in the GMS. Items 144 and 146 represent two of multiple logical ULP addresses. For each logical ULP address, there is a corresponding list, 147 and 148 of the host ULP addresses of the members of the logical message group. The logical message groups are not tied to specific ULP server processes, but are global with a GMS to all of the ULP server processes.

Control Functions

The control functions consist of connect, disconnect, create group, close group, join group, leave group, query groups, query group members, query group attributes. These control functions are implemented by a ULP send and receive control messages. The control functions are initiated by a host sending a ULP send control message to a GMS. These messages only allow a primary ULP destination address in the message and do no allow auxiliary addresses. The primary ULP address is interpreted as a control address space with a unique fixed address assigned to each of the control functions enumerated above. The contents of data in the payload supplies any arguments needed by the control function. Returned values from the control function are returned in a ULP receive control message that is addressed to the host that sent the original control message for which data is being returned. The detailed operation of these control functions is described below.

Connect

This control function allows a host to connect to a GMS. The destination ULP address in the message is a fixed address that indicates the connect function. The source ULP address and any data in the payload are ignored.

34

Upon receiving this message, the GMS control function, 136, creates a new host address and enters the host address in the host address map 136 along with the source TLP address from the TLP header of the message. Upon successful completion, the GMS control function responds with a receive

5   control ULP message addressed to the host along with a function code in the data portion of the payload that indicates successful host connection. The destination ULP address in the message is the ULP address assigned to the host. The host saves this and uses it for any future messages to the GMS. If there is an error, the control function returns a message to the host with a

10   function code in the data portion of the payload indicating failed host connection.

Disconnect

This function allows a host to disconnect from a GMS. The destination ULP address in the message is a fixed address that indicates the disconnect

15   function. The source ULP address is used to remove the host from membership in any implicit or logical groups prior to disconnecting. Any data in the payload is ignored. The GMS control function also removes the entry for the host from the host address map. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the

20   host along with a function code in the data portion of the payload that indicates successful host disconnection. The destination ULP address in the message is the ULP address assigned to the host. If there is an error, the control function returns a message to the host with a function code in the data portion of the payload indicating failed host disconnection.

35

Create implicit group

This function allows a host to create a new implicit message group and associated implicit ULP address and server process. The payload in the message may contain a single payload item whose data field holds attributes of the group. These attributes can be used to define any optional functions of the group. The destination ULP address in the message is a fixed address that indicates the create implicit group function. The GMS control function allocates a new implicit ULP address, adds it to the implicit ULP address list 138 and creates a new ULP server process 140. The host that sends this message is added to the membership list of the implicit group. This is done by adding the source ULP address in the message to the group membership list 142 in the ULP server process. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the host along with a function code in the data portion of the payload that indicates successful implicit group creation. The source ULP address in the payload is the ULP address assigned to the new implicit group. If there is an error, the control function returns a message to the host with a function code in the data portion of the payload indicating failed implicit group creation.

Create logical group

This function allows a host to create a new logical message group and associated logical ULP address. The payload in the message may contain a single payload item whose data field holds attributes of the group. These attributes can be used to define any optional functions of the group The destination ULP address in the message is a fixed address that indicates the create logical group function. The GMS control function allocates a new logical ULP address and adds it to the logical ULP address list 145. The host

that sends this message is added to the membership list of the logical group. This is done by adding the source ULP address in the message to the group membership list 147 for the new logical message group 144. Upon successful completion, the GMS control function responds with a receive control ULP

5   message addressed to the host along with a function code in the data portion of the payload that indicates successful logical group creation. The source ULP address in the payload is the ULP address assigned to the new logical group. If there is an error, the control function returns a message to the host with a function code in the data portion of the payload indicating failed implicit group

10   creation.

Join group

     This function allows a host to join an existing logical or implicit message group. The destination ULP address in the message is a fixed address that indicates the join group function. The data portion of the payload contains the

15   ULP address of the group that is to be joined. The GMS control function looks at this address and determines if it is an implicit or logical ULP address. If it is an implicit ULP address, the GMS control function finds the ULP server process selected by the address in the message payload and adds the source ULP host address from the message to the group membership list 142. If it is a

20   logical ULP address, the GMS control function finds the logical ULP address 144 selected by the address in the message payload and adds the source ULP host address from the message to the group membership list 147. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the host along with a function code in the

25   data portion of the payload that indicates successful group join. The source ULP address in the payload is the ULP address of the group that was joined. If

37

there is an error, the control function returns a message to the host with a function code in the data portion of the payload indicating failed implicit group creation.

Leave group

5   This function allows a host to leave an existing logical or implicit message group that it is a member of. The destination ULP address in the message is a fixed address that indicates the leave group function. The data portion of the payload contains the ULP address of the group that is to be left. The GMS control function looks at this address and determines if it is an implicit or

10   logical ULP address. If it is an implicit ULP address, the GMS control function finds the ULP server process selected by the address in the message payload and removes from the group membership list 142 the source ULP host address from the message. If the host is the last member of the group, the ULP server process is terminated and the implicit ULP address is de-allocated. If it

15   is a logical ULP address, the GMS control function finds the logical ULP address 144 selected by the address in the message payload and removes from the group membership list 147 the source ULP host address from the. If the host is the last member of the group, the ULP address is de-allocated. Upon successful completion, the GMS control function responds with a receive

20   control ULP message addressed to the host along with a function code in the data portion of the payload that indicates successful group leave. If there is an error, the control function returns a message to the host with a function code in the data portion of the payload indicating failed implicit group creation.

Query groups

25   This function allows a host to get a list of all implicit and logical message groups currently active on a GMS. The destination ULP address in the

38

message is a fixed address that indicates the query groups function. Any data portion of the payload is ignored. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the host along with a payload with multiple payload elements. The first payload element contains a function code indicating successful query groups. The source ULP address in the first payload element is ignored. Each of the subsequent payload elements contain a ULP group address in the source address field of the payload element that is one of the active group addresses on the GMS. There is no data field in these subsequent payload elements. If there is an error, the control function returns a message to the host with a function code in the data portion of a payload with a single payload element indicating failed query groups.

Query group members

This function allows a host to get a list of all hosts that are members of a message group. The destination ULP address in the message is a fixed address that indicates the query group members function. The data portion of the payload carries the address of the message group for the query. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the host along with a payload with multiple payload elements. The first payload element contains a function code indicating successful query group members. The source ULP address in the first payload element is ignored. Each of the subsequent payload elements contain a ULP host address in the source address field of the payload element that is one of the active group addresses on the GMS. There is no data field in these subsequent payload elements. If there is an error, the control function

returns a message to the host with a function code in the data portion of a payload with a single payload element indicating failed query group members.

Query group attributes

This function allows a host to get a list of the attributes of a message group. The destination ULP address in the message is a fixed address that indicates the query group attributes function. The data portion of the payload carries the address of the message group for the query. Upon successful completion, the GMS control function responds with a receive control ULP message addressed to the host along with a payload with a two payload elements. The first payload element contains a function code indicating successful query group members. The second payload element contains the attributes of the message group. If there is an error, the control function returns a message to the host with a function code in the data portion of a payload with a single payload element indicating failed query group attributes.

Send Message Operation

In order to fully understand the operations of the send message function, a number of individual cases are worth considering.

Single implicit destination

The most simple case is a send message to a single implicit ULP address. In all send message datagrams, the destination ULP address 125 must be an implicit ULP address. In this case of a single implicit destination, this is the only destination address in the datagram. The auxiliary address count 126 is zero and there are no auxiliary destination addresses 127 or 128. The payload consists of a message count 116 of one, the ULP of the host sending the message in the source ULP address 117 and the data length 118 and data 119.

Send message datagrams may only have a single payload item so their message count field 116 must always be one.

The host sends the send message onto the network with a TLP header addressing the datagram to the GMS that is the selected target of the message. The GMS receives the message and the GMS control function 136 determines that it is a send message datagram and looks up the implicit destination address in its implicit ULP address list 138. If the address does not exist, an error message is returned to the sending host with a ULP receive message datagram. If the address is valid, the GMS control function removes the TLP header from the datagram and sends the ULP portion to the ULP server process corresponding to the destination implicit ULP address. Assume for discussion that this is the ULP server process 140. The ULP server process 140 will extract the single payload item from the message 117, 118 and 119 and place the payload item in each of the message queues 143. There will be one message queue for each member of the message group served by the ULP server process 140. The members of the group will have their host ULP addresses listed in the host address list 142. Each message queue in a ULP server process will fill with payload items that are targeted at particular destination hosts. The mechanisms by which payload items are removed from the queues and sent to the hosts will be described later.

Auxiliary unicast destination

In this case in addition to an implicit destination 125, there is also a single auxiliary address 127 in the datagram. The auxiliary address count 126 is one and the auxiliary destination addresses 127 is a unicast host ULP address. The payload consists of a message count 116 of one, the ULP of the host sending

the message in the source ULP address 117 and the data length 118 and data 119.

The host sends the send message onto the network with a TLP header addressing the datagram to the GMS that is the selected target of the message. The GMS receives the message and the GMS control function 136 determines that it is a send message datagram and looks up the implicit destination address in its implicit ULP address list 138 and the unicast host ULP auxiliary address in the host address map 137. If either of addresses does not exist, an error message is returned to the sending host with a ULP receive message datagram. If the addresses are valid, the GMS control function removes the TLP header from the datagram and sends the ULP portion to the ULP server process corresponding to the destination implicit ULP address. Assume for discussion that this is the ULP server process 140. The ULP server process extracts the auxiliary ULP address from the message and determines from the address that it is a unicast host ULP address. The server process then checks to see if this address is a member of the message group defined by the host address list 142. If it is not, no further action is taken and the payload item in the message is not placed in any of the message queues 143. If the host address is in the message group, the payload item in the message is placed in the single message queue corresponding to that host. The net effect is that the ULP server process has performed a set intersection operation on the members of the message group selected by the implicit ULP destination address and defined by the group membership list 142 with the members of the set of hosts defined by the auxiliary address. The payload item is them sent only to the hosts that are members of this set intersection.

Auxiliary logical destination

In this case in addition to an implicit destination 125, there is also a single auxiliary address 127 in the datagram. The auxiliary address count 126 is one and the auxiliary destination addresses 127 is a logical ULP address. The

5   payload consists of a message count 116 of one, the ULP of the host sending the message in the source ULP address 117 and the data length 118 and data 119.

The host sends the send message onto the network with a TLP header addressing the datagram to the GMS that is the selected target of the message.

10   The GMS receives the message and the GMS control function 136 determines that it is a send message datagram and looks up the implicit destination address in its implicit ULP address list 138 and the logical ULP auxiliary address in list of logical ULP addresses 145. If either of addresses does not exist, an error message is returned to the sending host with a ULP receive message datagram.

15   If the addresses are valid, the GMS control function removes the TLP header from the datagram and sends the ULP portion to the ULP server process corresponding to the destination implicit ULP address. Assume for discussion that this is the ULP server process 140. The ULP server process extracts the auxiliary ULP address from the message and determines from the address that

20   it is a logical ULP address. Assume for this example that this logical ULP address is the logical address 144. The server process fetches the group membership list 147 corresponding to the logical address and performs a set intersection operation with the group membership list 142 of the server process. If there are no members of this set intersection, no further action is

25   taken and the payload item in the message is not placed in any of the message queues 143. If there are members of the set intersection operation, the payload

item in the message is placed in the queues corresponding to the hosts that are members of the set intersection.

Multiple auxiliary addresses with logical operations

In its most sophisticated form, a send message can perform set operations
between the implicit message group of the ULP server process and multiple
logical and unicast ULP addresses. This is done by placing multiple auxiliary
destination ULP addresses in the message with logical operators imbedded in
the address list. The address count 126 holds a count of the total auxiliary
addresses in the address list 127 and 128. The auxiliary addresses are a mix of
logical ULP addresses and unicast host ULP addresses. Two logical ULP
addresses in the ULP address space are assigned the role of specifying set
operations to be performed between the logical message groups and unicast
host addresses in the message list. They are specially assigned addresses for
the functions set intersection, set union. A third logical address is used to
indicate set complement. The payload consists of a message count 116 of one,
the ULP of the host sending the message in the source ULP address 117 and
the data length 118 and data 119.

The host sends the send message onto the network with a TLP header
addressing the datagram to the GMS that is the selected target of the message.
The GMS receives the message and the GMS control function 136 determines
that it is a send message datagram and looks up the implicit ULP message in
the implicit ULP address list 138 and all of the addresses in the address list
either in the host ULP address map 137 or in the logical ULP address list 145
as appropriate. If any of addresses does not exist, an error message is returned
to the sending host with a ULP receive message datagram. If the addresses are
valid, the GMS control function removes the TLP header from the datagram

and sends the ULP portion to the ULP server process corresponding to the destination implicit ULP address. Assume for discussion that this is the ULP server process 140. The ULP server process extracts the auxiliary ULP address list from the message and scans it from beginning to end. The scanning

5   and processing of the set operators is done in post-fix fashion. This means that arguments are read followed by an operator that is then applied to the arguments. The result of the operator becomes the first argument of the next operation. Therefore at the start of scanning two addresses are read from the address list. The next address will be an operator that is applied to the

10   arguments and the result of this operator is the first argument to be used by the next operator. From then on a single address is read from the address list followed by a logical ULP address which is operator on the two arguments consisting of the new argument and the results of the last operator. The logical address used to indicate set complement is not a set operator, by an argument

15   qualifier since it can precede any address in the address list. The meaning of the set complement argument qualifier is relative to the group membership of implicit group address in the send message. If the set complement qualifier precedes a unicast host address which is not a member of the message group selected by the implicit ULP address in the send message, the effective

20   argument is the set of all hosts that are members of the implicit message group. If the set complement qualifier precedes a unicast host address which is a member of the message group selected by the implicit ULP address in the send message, the effective argument is the set of all hosts that are members of the implicit message group except for the original unicast host address qualified by

25   the complement function. If the set complement qualifier precedes a logical ULP address the effective argument is the set of all hosts that are members of

the implicit message group specified by the send message except hosts that are members of the logical message group preceded by the set complement modifier. Once the entire address list has been processed to a single result set of hosts, a set intersection operation is performed on this set and the set of

5   members of the implicit message group 142 defined by the implicit address in the send message. If there are no members of this set intersection, no further action is taken and the payload item in the message is not placed in any of the message queues 143. If there are members of the set intersection operation, the payload item in the message is placed in the queues corresponding to the

10   hosts that are members of the set intersection.

Message Delivery and Aggregation

Once messages are entered into the message queues in the ULP server processes, there are a variety of ways that they can ultimately be delivered to the targeted hosts. In the invention, the delivery method is set on a per-ULP

15   server process basis by attributes that are provided at the time that an implicit ULP message group and server process are created. It is important during the description of these methods to keep in mind that the invention is intended to provide an efficient means for a group of hosts to send messages to each other at a rapid rate during the implementation of a networked interactive

20   application. Also assumed in the following description is that the GMS performs echo suppression when a host sends a message to a group that it belongs to. This means that the host will not receive a copy of its own message to the group either as a single un-aggregated message or as a payload item in an aggregated message. This is controlled by a ULP server process attribute

25   that can be changed to stop echo suppression, but echo suppression is the default.

Immediate Delivery

The most simple delivery method is to immediately deliver the payload items to their targeted hosts as soon as they are placed in the message queues. Each payload item in a message queue will contain a ULP source address, a data length and the data to be sent. To implement immediate delivery, the ULP server process will remove a payload item from a message queue for a particular host 143. The host address for this host will be obtained from the group membership list 142. The payload item and the destination host address will be sent to the GMS control function 136 where it will be used to create a ULP receive message sent to the destination host. The GMS control function 136 will use the destination ULP host address to look up the TLP address of the host from the host address map 137. This will be used to create a TLP header for the message 123. The ULP message type 124 will be ULP receive, the destination ULP address 125 will be the destination host, the address count will be 0 and there will be no auxiliary addresses. The payload in this case will have a message count 116 of 1 and the payload item comprised of fields 117, 118, and 119 will be the payload element taken from the message queue.

Immediate delivery is useful when the message rate between a group of hosts is low. Consider four hosts that are members of an implicit message group where each member of the group sends a message to every other member of the group at a fixed rate. With immediate delivery, each host will send three messages to the other members of the group and receive three messages from the other members of the group at the fixed rate. This is acceptable is the size of the group is small and the message rate is low. However, it is obvious that total message rate is the product of the underlying message rate and the total number of members of the group minus one. Clearly

this will result in unacceptably high message rates for large groups and highly interactive message rates. A group of 20 members that had an underlying message rate of 10 messages per second would yield a total message rate at each host of 190 messages sent and 190 messages received every second. This

5   message rate will be unsupportable over a conventional dial-up connection to a conventional wide area network such as the internet.

Aggregation

         A key concept in the present invention is the aggregation of multiple messages in a message queue into a single ULP receive message to a host that

10   contains multiple payload items in the payload. The ULP server process 140 removes payload items from a message queue 143 for a host and accumulates them in an aggregation buffer 149. The aggregation buffer has buffer areas for each host for which there is a message queue. These individual host areas within the aggregation buffer are called host aggregation buffers. The start and

15   end of this aggregation period can be controlled in a number of ways that will be described in the next sections. At the end of the aggregation period, the each host aggregation buffer may hold multiple payload items. The host aggregation buffer will hold a message count of the payload items followed by the multiple payload items. The contents of a host aggregation buffer along

20   with the ULP host address of the corresponding host are sent to the GMS control function 136 where it will be used to create a ULP receive message sent to the destination host. The GMS control function 136 will use the destination ULP host address to look up the TLP address of the host from the host address map 137. This will be used to create a TLP header for the

25   message 123. The ULP message type 124 will be ULP receive, the destination ULP address 125 will be the destination host, the address count will be 0 and

there will be no auxiliary addresses. The payload in this case will have a message count 116 set by the message count value from the host aggregation buffer. The payload will contain all of the payload items from the host aggregation buffer.

5   The effect of aggregation will be to greatly reduce the total message rate received by the hosts. A single message to a host will be able to carry multiple payload items received from the other hosts during the aggregation period. This fits very well the interactive applications of this invention where groups of hosts will be sending messages to all the other hosts in the group at a periodic

10   rate. Aggregation will be very effective in collecting together all of the messages from all of the other hosts into a single message for each member of the group. The reduces processing at each receiving host since a single message will be received rather than many separate messages. Aggregation will also reduce the total data rate to the hosts since aggregation eliminates the

15   need for separate message headers for each payload item. The savings will be significant for small payload items since there will be only one message header comprising fields 123, 124 and 125 for multiple payload items. In cases where a group of hosts are sending messages to the group at a periodic rate, it is often the case in many interactive applications that the data being sent by each host

20   to the group is very similar to the messages sent by the other hosts. This affords the opportunity within an aggregated payload of multiple payload items to apply a data compression method across the multiple data elements of the payload elements. A wide variety of known data compression methods will lend themselves to this application. The first data element in the first payload

25   item can be sent in uncompressed form with each subsequent data element being compressed using some form of difference coding method. A variety of

known data compression methods use the concept of a predictor with differences from the predicted value being encoded. The first data element in an aggregated payload can be used as this predictor with the subsequent data elements coded using such a data compression method. These conventional data compression methods do not assume any knowledge of the internal structure or function of portions of a data element to compress. It is also possible to make use of application specific coding techniques that take advantage of such knowledge to potentially achieve much higher coding efficiency.

Server Isochronous

One method by which the aggregation time period can be defined is called Server Isochronous or SI. In this method, A ULP Server Process defines a uniform time base for defining the aggregation time period. This time base is defined by three parameters: the time period, the aggregation offset and the transmit offset. These parameters are set by the attributes provided in the create implicit group control function at the time the implicit group and the ULP server process are created. The time period is a fixed time interval during which the ULP server process will accumulate messages in the message queues, aggregate the messages in the queues and send the aggregated messages to the targeted hosts. The aggregation offset defines the point after the start of the time period after which arriving messages will be stored in the message queues for delivery in the next time period. Therefore, at the aggregation offset after the start of the time period, a snapshot will be taken of all of the messages in each message queue. New messages will continue to arrive and be entered into the queues after the aggregation offset. Only those messages in the queues before the aggregation offset point will be aggregated

into outbound messages. The resulting aggregated messages will then be sent to their targeted hosts at the point in time which is the transmit offset after the start of the time period. The result is that messages arrive continuously and are stored in the message queues. Once per time period the are aggregated into

5   single messages to each host which is the target of messages and once per time period these aggregated messages are sent to the hosts.

Another embodiment of the SI method is to allow the ULP server process to dynamically vary the time period based on some criteria such as the received message rates, and/or received data rate. The ULP server could use a function

10 to define the aggregation period based on the number of messages received per second or the total number of payload bytes received per second. One reasonable function would be to shorten the aggregation period as the rate or received messages or data rate of the received payloads increased. This would tend to keep the size of the outbound messages from growing too much as

15 received messages and/or received data rate grew. Other possible functions could be used that varied the aggregation period based on received message rates, received payload data rates or other parameters available to the ULP server process.

Host Synchronous

20 The host synchronous or HS method of defining the aggregation time period allows the definition of a flexible time period that is controlled by the hosts. It is based on the concept of a turn which is a host sending a message to one or more members of the implicit message group which is operating is HS mode. Once every host in the message group has taken a turn, the aggregation

25 period ends. A snapshot of the contents of the message queues is taken, the contents of each of the queues is aggregated and the aggregated messages are

sent to the hosts targeted by each message queue. A refinement to this technique qualifies which of the three ULP send message types to the group constitute a host turn: a send only to the implicit address of the group, a send to a unicast host address within the group or a send to a logical ULP address which shares members with the group. The attributes of the group not only will define HS aggregation, but one or more ULP send message types that will be considered a host turn. A further refinement sets the total number of turns that a host can take in a single aggregation time period. The default will be one turn, but multiple turns can be allowed. If a host attempts to take more turns than allowed, the messages are ignored.

This aggregation technique has the additional benefit of causing the hosts which are member of an HS implicit message group to have their processing functions synchronized when they are executing the same interactive application. Many networked interactive applications are based on a simple overall three step operational model: wait for messages from other hosts, process the messages and the local users inputs to update the local application, send messages to the other hosts. This basic application loop is repeated at a rate fast enough to provide an interactive experience such as 5 to 30 times per second. It is desirable to keep such applications synchronized so that the states of the applications is consistent on the different host machines. When such applications communicate using the HS model of the present invention their operations will become naturally synchronized. The HS ULP server process will wait until all of the members of the message group has completed their turns and sent a message to the group before sending the aggregated messages to the members of the group. This will cause the applications on the hosts to wait until they have received the aggregated messages. They will all then start

processing these messages along with the local user inputs. Even if they perform their processing at different speeds and send their next messages to the group at different times, the HS ULP server will wait until all have completed their processing and reported in with a message to the group. This will keep all

5   of the host applications synchronized in that every host will be at the same application loop iteration as all of the others. This will keep the application state consistent on all of the hosts. Only network propagation delays from the GMS to the hosts and different processing speeds of the hosts will cause the start and completion of their processing to begin at different times. It is not a

10   requirement in networked applications to keep all of the hosts precisely synchronized, only that that application state is consistent. The HS method provides a natural way to do this in the context of the present invention.

Preferred Embodiment

       The detailed description of the invention has described a datagram

15   implementation of the invention as the best way to explain the invention. The preferred embodiment of the invention is as follows.

       In the preferred embodiment, the wide area network is the Internet and the TLP protocol is TCP/IP. The GMS is a general purpose computer system connected to the Internet and the hosts are personal computers connected to

20   the Internet.

       TCP/IP provides an number of advantages that provide for a more efficient applications interface on the hosts 151. TCP/IP supports the concept of source and destination port numbers in its header. The ULP can make use of the port numbers to identify source and destination ULP connections. Most ULP send

25   messages will be from hosts to a implicit ULP group addresses and most ULP receive messages will be from the implicit ULP addresses to the ULP host

addresses. All of these and the ULP message type field can represented by source and destination port addresses within the TCP/IP header. This means that for most ULP messages, the ULP message encapsulated within the TCP/IP message need only contain the payload. There is the slight complication of the aggregated ULP receive messages sent from a ULP server process to a hosts. Here the destination port will be the host the source port will be for the implicit ULP group address and the payload will still contain the source host ULP addresses in each the payload items.

TCP/IP also supports header compression for low speed dial-up lines which is also important in this application. See RFC 1144. TCP/IP is a connection oriented protocol which provides reliable end-to-end transport. It handles re-transmission on errors and fragmentation and reassembly of data transparently to upper level protocols. Header compression allows much of the TCP/IP header to be omitted with each packet to be replaced by a small connection identifier. This connection ID will uniquely define a connection consisting of a source and destination IP address and source and destination TCP/IP port numbers.

At the interface to the application on the hosts, the preferred embodiment of the ULP is as a session layer protocol. In the preferred embodiment the application on a host opens a session with a ULP server process. This session is identified with a unique session ID on the host. The host application then sends data to the ULP host interface 151 tagged with this session ID. The session ID defines a host and implicit ULP pair including the TCP/IP TLP address of the GMS server that is running the particular ULP server process for the implicit ULP address. By binding the transport address of the GMS of a ULP server process to the session ID, we can transparently to the application

support multiple group messaging servers on the network and a single host can have multiple active sessions with different physical group messaging servers. This avoids any address space collision problems that could arise from the fact that the ULP address space is unique to each GMS.

Alternate Embodiments

One possible extension to the invention is to extend the ULP to support a common synchronized time base on the GMS and the hosts that are connected to it. This would be most interesting in context of the SI message aggregation mode. The SI time base on the GMS could be replicated on all of the hosts and all of the hosts and the GMS could lock these time bases together. There are known methods to synchronize time bases on multiple computer systems. One such method is called NTP.

Another extension to the invention is to define ULP server processes that perform specific application specific processing on the contents of the messages that are received. A variety of different application specific processing functions can be defined and implemented. A particular function would be selected by attributes provided in the create implicit group function. These functions could process the data in the message payloads and replace the data elements in the payloads with processed results. Separately, or in combination with processing the message payloads, the processing could store either raw message payload data in the application specific state storage area or could store processed results.

Clearly, the host system need not be personal computers, but could also be dedicated game consoles or television set top boxes or any other device with a programmable controller capable of implementing the ULP protocol.

The wide area network used to transport the ULP protocol need not be the Internet or based on IP. Other networks with some means for wide area packet or datagram transport are possible including ATM networks or a digital cable television network.

5   The invention now being fully described, it will be apparent to one of ordinary skill in the art that any changes and modifications can be made thereto without departing from the spirit or scope of the invention as set forth herein. Accordingly, the present invention is to be limited solely by the scope of the appended claims.